**Heart Failure Clinical Records**

**Tareas:**

1. **Analizar y Comprender el dataset atraves de un analisis exploratorio de datos.**

Conclusiones del EDA(EDA\_Personal\_Project.ipynb):

1. Transformacion de datos: despues de analizar los datos decidi mantenerlos exactamente como se estaban mostrando, ya que no considero que se necesite cambiar para el modelo, la mayoria de los valores son enteros o flotantes, y como el modelo es uno de clasificacion, nos serviran estos valores como se encuentran.
2. Limpieza de datos: al hacer el analisis de cuantos datos estaban como vacios, descubri que no habia valores vacios asi que en este paso no hubo necesidad de eliminar valores.
3. Deteccion de outliers:

Existen outliers en 5 features diferentes: creatinine\_phosphokinase, ejection\_fraction, platelets, serum\_creatinine y serum\_sodium.

|  |  |
| --- | --- |
| Feature | Total de Outliers |
| creatinine\_phosphokinase | 29 |
| ejection\_fraction | 2 |
| Platelets | 21 |
| serum\_creatinine | 29 |
| serum\_sodium | 4 |

De los cuales 12 records tienen una union de 2 features donde existen outliers:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **creatinine\_phosphokinase** | **diabetes** | **ejection\_fraction** | **high\_blood\_pressure** | **platelets** | **serum\_creatinine** | **serum\_sodium** |
| 160 | 1 | 20 | 0 | 327000 | 2.7 | 116 |
| 2656 | 1 | 30 | 0 | 305000 | 2.3 | 137 |
| 3964 | 1 | 62 | 0 | 263358.03 | 6.8 | 146 |
| 102 | 0 | 60 | 0 | 507000 | 3.2 | 138 |
| 1082 | 1 | 45 | 0 | 250000 | 6.1 | 131 |
| 2334 | 1 | 35 | 0 | 75000 | 0.9 | 142 |
| 66 | 1 | 20 | 0 | 70000 | 2.4 | 134 |
| 1211 | 1 | 35 | 0 | 263358.03 | 1.8 | 113 |
| 1767 | 0 | 45 | 0 | 73000 | 0.7 | 137 |
| 427 | 0 | 70 | 1 | 151000 | 9 | 137 |
| 582 | 0 | 40 | 0 | 51000 | 2.7 | 136 |
| 2060 | 1 | 60 | 0 | 742000 | 0.8 | 138 |

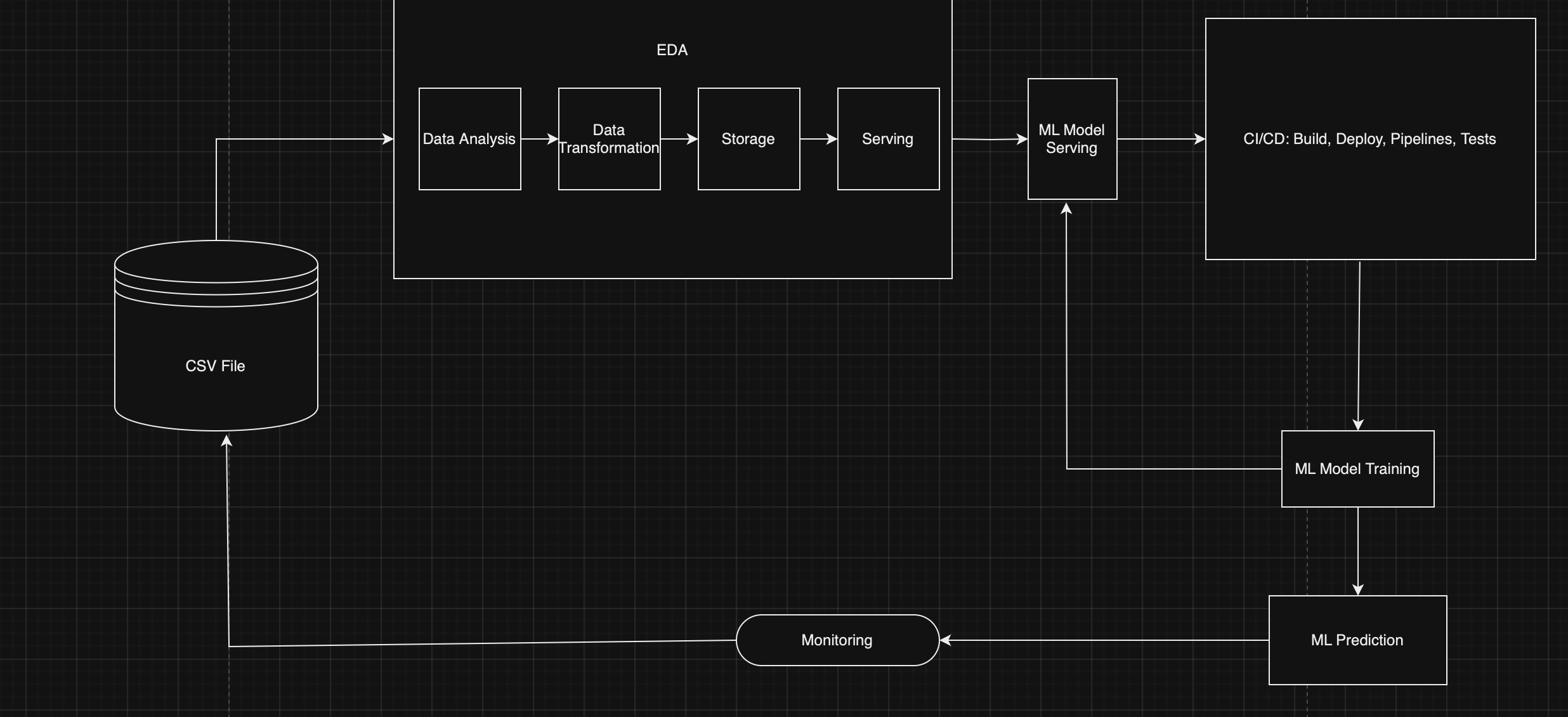
Al hacer la grafica de calor entre los 2 datasets (original y sin outliers hubo una correlacion mayor entre algunas features, como ejection\_fraction, serum\_creatinine, sin embargo tambien veo que en cuanto al feature de sex/death\_event antes de eliminar outliers era la correlacion negativa y sin outliers es positiva. Las diferencias son minimas en los valores.

Por ultimo agregue una comparativa de los datos antes y despues de los outliers, comparando algunas features entre si, y en los pictogramas se ven los datos mas limpios sin los outliers que tenian varios features con outliers.

1. **Determinar la pregunta que deseas abordar con un modelo de aprendizaje automatico:**  
   Se puede deteminar si un paciente tendra una falla en el corazon ?
2. **Identificar por que se necesita una estrategia de MLOps para este dataset:**

Despues de realizar el EDA se encuentra uan correlacion entre varias features las cuales se pueden usar para crear un modelo que haga una predicion sobre si el paciente tendra una falla en el corazon usando como informacion algunas caracteristicas de la sangre, y las funciones del corazon como bombeo de sangre/presion.

1. **Diseñar la arquitectura del pipeline, para esta nueva iniciativa de aprendizaje automatico:**
2. El primer paso es el obtener los datos del dataset.
3. Despues en el EDA: transformamos los datos, hacemos un analisis en base a las features y limpiamos el dataset y entregamos los datos ya listos para usarse en el modelo.
4. El paso 3 seria la creacion del modelo.
5. El siguiente paso involucra el despliegue del modelo asi como el seteo del ambiente.
6. El siguiente paso es el entrenamiento del modelo ya con los valores de prod
7. Ultimo paso es la prediccion y monitoreo del modelo, incluye el mantenimiento del modelo.



1. **Crear un modelo:**

Conclusiones

1. Github: https://github.com/rocNajLpz/failureheart-mlops/models/Modelo.ipynb
2. Agregue 3 modelos: Logistic Regression, Decision Tree y Random Forest y use 2 features que tenian major correlacion en el EDA: serum sodium y ejection fraction, los resultados de la presicion muestran que el random forest tiene un 91 %, que seria el mejor de los 3 modelos.
3. **Configurar la estructura correcta del modelo con la idea de dejarlo listo para el despliegue.**

**Organice el proyecto en 6 folders diferentes:**

* **Data: aquí se encuentran los datasets, hay 2 folders raw y processed. Raw tiene el dataset original y processed tiene el dataset con el que se va trabajando en los diferentes stages.**
* **Documentation: aquí el archivo que se esta leyendo, viene la informacion de cada una de las tareas entregables.**
* **Models: aquí tengo el archivo model.joblib que se genera el stage cuando se procesan los datos y se corre el proyecto**
* **Notebooks: aquí tengo los archivos notebook iniciales que se crearon en la fase inicial, incluyen el EDA y el modelo.**
* **Reports: Aquí el pipeline esta generando un archivo donde guarda las metricas del modelo.**
* **Src: esta carpeta contiene toda la parte del codigo del proyecto.**
  + **App: contiene los archivos referentes a la configuracion de dockers y el main que ejecutara cuando docker este listo**
  + **Stages: aquí se tienen todos los archivos .py de los stages que se configuran en el archivo .yaml**
  + **Utils: aquí tiene el archivo logs que contiene el codigo para generar logs**

1. **Crear nuevas versiones del modelo, que incluyan cambios en las características o ajustes de hiper parámetros para la reproducibilidad y el seguimiento experimental. Agregar estrategias reproducibles y de prueba que se aplicarán a tu pipeline de aprendizaje automático.**

**Dentro del archivo de dvc.lock podemos ver las diferentes version que se han generando con los cambios en el ambiente local del modelo.**

1. **Implementar un modelo en su entorno local utilizando contenedores y planificar el reentrenamiento, el drift, el redeploy, el escalado y el monitoreo. Orquestar y registrar el pipeline de aprendizaje automático, y se realizar el despliegue inicial.**

*Para este paso logre hacer el despliegue inicial, el pipeline y cree el archivo de dockers, sin embargo, no pude hacer funcionar docker en mi computadora de Windows, lo que logre fue crear la imagen dentro de github creando un action*

1. **Demostrar estrategias de prueba y versionamiento. Evaluar posibles derivas que puedan ocurrir e incluir tareas de monitoreo y pruebas adicionales para abordarlas. Faltante**

Preguntas del Entregable

Recuerda ir contestando las siguientes preguntas:

1. ¿Qué elementos matemáticos se consideraron en esta decisión? Clasificacion, correlacion, f1

2. ¿Cómo se integrarán nuevos datos?NA

3. ¿Cómo se medirá el drifting?no pude lograr crear el drift, cree el archivo de dockers pero no pude lograr deployar en cloud

4. ¿Se considera la prueba en el desarrollo del pipeline?

5. Agrega tus implementaciones de código

6. Agrega tus diagramas/elementos gráficos para ilustrar y facilitar la comprensión de la solución general.